

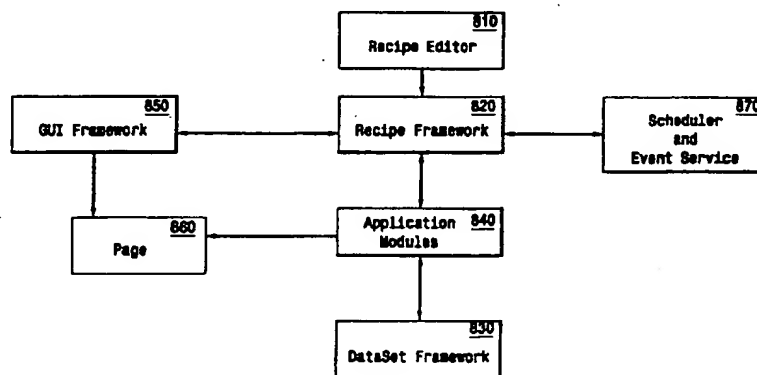


A

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06F 17/00, 17/50		A1	(11) International Publication Number: WO 99/22310
			(43) International Publication Date: 6 May 1999 (06.05.99)
(21) International Application Number: PCT/US98/22735		(81) Designated States: JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).	
(22) International Filing Date: 27 October 1998 (27.10.98)		Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>	
(30) Priority Data:			
08/958,288	27 October 1997 (27.10.97) US		
08/958,780	27 October 1997 (27.10.97) US		
(71) Applicant: KLA-TENCOR CORPORATION [US/US]; 160 Rio Robles, San Jose, CA 95134 (US).			
(72) Inventors: HARDIKAR, Manoj; 20727 Russell Lane, Saratoga, CA 95070 (US). ZHOU, Steve; 21073 Grenola Drive, Cupertino, CA 95014 (US). SHIFLETT, Richard; 866 Willow Glen Way, San Jose, CA 95125 (US). KULKARNI, Ashok; 1149 Valley Quail Circle, San Jose, CA 95120 (US).			
(74) Agent: HITE, J., Eppa; Carr & Ferrell LLP, Suite 200, 2225 E. Bayshore Road, Palo Alto, CA 94303 (US).			

(54) Title: **SOFTWARE SYSTEM AND METHOD FOR EXTENDING CLASSIFICATIONS AND ATTRIBUTES IN PRODUCTION ANALYSIS**



(57) Abstract

A software system is used for extending classification attributes in the analysis of production data. The Recipe Editor (810) permits the user to create a Recipe Framework (820). The recipe is applied to the DataSet Framework (830) through the Application Modules (840), and with the Graphic User Interface (GUI) Framework (850), a Page (860) is produced in the Window format. The Recipe Framework (820) also interfaces with the Scheduler and Event Service (870), providing automatic execution and publication of production data analyses at future specified times. The software operates in an object-oriented Windows environment with increased flexibility because it permits the user to add defect classifications by dragging and dropping with a computer mouse. When a cluster classification is executed, a production map is filtered to remove the non-defect data according to selected classification; the defect attribute is then readily visible.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

**SOFTWARE SYSTEM AND METHOD FOR EXTENDING
CLASSIFICATIONS AND ATTRIBUTES IN PRODUCTION
ANALYSIS**

RELATED APPLICATIONS

This application is related to a co-pending PCT Patent Application corresponding to U.S. Patent Application Serial No. 08/958,780, filed on October 27, 1997, and entitled, "Software System and Method for Graphically Building Customized Recipe Flowcharts," which is hereby incorporated by reference.

BACKGROUND OF THE INVENTION

1. **Field of Invention**

This invention relates to computer software for quality control applications of semiconductor production lines, particularly, a software package for extending namespace classifications in the analysis of production data.

2. **Description of the Prior Art**

Software packages are increasingly being written for Windows® formats so that a user may interact with the program through symbols on the computer screen via mouse clicks or arrow keys. Such programs are typically written in Visual C++ or Visual Basic programming language, and permit the programmer to call on generalized programming objects to display selectable data input fields. These programs are more user-friendly and afford greater flexibility because the user can choose to select from any one of a large number of program functions based upon displayed visual cues.

Object oriented programming permits a programmer to call upon a particular functional subroutine in Windows® which is modular in design and operable across a large number of different hardware systems. One primary

advantage of Windows® is that a great number of programming objects are included in the basic system. Examples of visual objects are the classes of buttons, list boxes, dialog boxes, toolbars, scroll bars, menu bars and title bars. Mouse functions are also enabled. A programmer working within a
5 Windows® environment has access to all of these objects, adding considerable power to his programming abilities.

Quality control has historically relied upon automated computer systems. The automated production line replaces the manual laborer and automated quality control eliminates the inspectors. One important
10 consumer of quality control is the semiconductor industry. Microscopic defects in a semiconductor wafer have major implications on the cost and yield of a semiconductor production process. Because semiconductor technology is so sensitive to defects, isolating their precise source in the production process facilitates great advancement in the industry.

15 Quality control software packages are currently available for the semiconductor industry, but the programs are generally written on a project-specific basis. These programs are written to include a large number of analytical utilities around a core system but these utilities are essentially fixed in number and the program is difficult to modify. Thus, these
20 packages are expensive and lacking in flexibility.

One method of measuring the number and location of defects in a semiconductor chip involves filtering or erasing the non-defect data from the production data to leave only the defect data. A map of actual production data either may be compared to a map of another set of actual production
25 data, or to a map of an ideal set of production data stored in the computer memory. The defect data is then analyzed according to number, density, location, type and step in the production process, with the goal of correcting the production process itself. The defect data may give clues about how to increase the yield and decrease the cost of the product.

30 A variety of prior art data analysis systems and methods have been disclosed for industrial process control of semiconductor manufacturing. Among these are U.S. Patent No. 5,226,118 to *Baker et al.* This patent discloses a system for making charts. The charts are displayed as a gallery

of cells where each cell is a unit of a two dimensional array. Each cell in the chart may be defined as a mathematical function of one or more other cells in a manner similar to a spreadsheet program such as Microsoft Excel.[®]

Other related patents are U.S. Patent Nos. 4,967,381; 4,951,190;
5 4,873,623; 4,843,538; 4,805,089; and 4,679,137. These patents disclose a complex multilevel set of data structures. While the control parameters can be changed to some extent by the engineer, the set of parameters is predefined by the software package. Moreover, not all parameters can be changed, and the extent to which a given operator may change them
10 depends upon the operator's level of authorization. These systems are somewhat primitive because they do not recognize the flexibility afforded by object-oriented visual programming.

SUMMARY OF THE INVENTION

The present invention concerns a software system and method for extending cluster classifications for an analytical routine of production data. The user selects a new classification from a list box and adds the new
5 classification to a table namespace of old classifications by dragging and dropping with a computer mouse. The new cluster classification is a namespace with attributes for executing the analytical routine of this cluster classification.

The set of classifications in the table namespace is executable for
10 graphically displaying analytical results in a Windows® environment. The particular classifications are clusters of defect data in a production map. Examples of features identified by these cluster classifications are scratches, particles, pinholes and blowouts. Execution of a particular cluster
15 classification filters the production data so that only a map of the particular defect of the cluster classification remains. The attributes of the classification cluster may also retrieve an analytical routine across a local network or the Internet so that new cluster classifications may be added from other fabrication plants.

Charts, such as Type Pareto or bar graphs, are produced by the
20 analytical routines of the cluster classifications through their attributes to analyze data from production runs based upon production functions. The software permits the user to color code the defects in each cluster classification for easy identification. A user may move from one page of a filtered production map to another page, or place two pages side by side for
25 comparison in order to look for patterns establishing where the production defects arise. Thus, the pattern of defects on a given map may be visualized more easily.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a computer screen for specifying data analysis parameters;

Figure 2A shows a bar chart histogram produced by the software;

5 Figure 2B shows a calendar produced by the software;

Figure 3 illustrates a display window where two or more bar charts are placed side by side for comparison;

Figure 4A shows a zoom feature on the software package for viewing production data;

10 Figure 4B shows a production map after filtering;

Figure 5 shows the display window of a flowchart created by the software package;

Figure 6A illustrates the overall system architecture;

15 Figures 6B and 6C illustrate the interfaces of the middle layer architecture;

Figure 7 illustrates the operation of the higher level software;

Figure 8A a format for a namespace;

Figure 8B shows a list of cluster classifications for analyzing defect data and a list box for adding cluster classifications;

20 Figure 8C is a block diagram illustrating attributes acting across the Internet; and

Figure 8D is a flowchart showing attributes acting across the Internet.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

1. Chart Generation

Figure 1 illustrates a computer screen 100 having one or more dialog boxes 101, 102, 103 for specifying data analysis parameters. Drop-down list boxes 101, 102 indicate which parameters are to be placed on the x- and y-axes of the bar chart. The parameters are selected with a computer mouse from a drop-down menu or similar type of dialog box 101, 102. The user clicks on any of the parameters in the menu of the dialog box 101 or 102 to select the data analysis parameters on the x- and y- axes.

Preferably, the dialog box 101, 102 is a drop down list box with a vertical scroll bar. The user causes the list box 101, 102 to drop down by clicking on the arrow box 111, 112 beside the list box 101, 102. The user clicks on one of a collection of text strings representing production variables and production functions within a columnar list within the list box rectangle. When the desired text string is found within the list box 101, 102, the user clicks on this text string which causes it to appear in the selection line of the list box 101, 102.

One feature of the present invention is the recording of the selected production variables and production functions in a macro feature. The computer thus remembers or learns the selection of variables and functions selected by the user, and has the capability of replaying the particular variables and functions for a new data set. The analytical system and method therefore differs from the prior art where new production parameters must be specified each time an analysis is to be performed, or where the user is restricted to choosing from only a specified set of preprogrammed analytical schemes. The macro feature stores the particular set commands relating to the production parameters that the user has previously selected from the dialog boxes and drop down list boxes 101, 102.

Parameters applicable to monitoring production defects might include number on the y-axis and type on the x-axis. A Type Pareto Chart is then created and displayed. For example, the number of defects in a

semiconductor assembly line is selected as the parameter on the y-axis. The x-axis parameter is selected from a list including defect size or dates of production.

Figure 2A shows a Type Pareto Chart 200 of the number of defects by defect class. An appropriate title bar 210 may be found above the window, in this case reading "Type Pareto Chart." At the bottom of the screen are tabs 221, 222, 223, 224 indicating a number of pages that are immediately accessible to the user. In Figure 2A, the pages are labeled "1 Type Pareto," "2 Size Histogram," "3 Wafermap," and "4 Wafermap Gallery." By clicking on these tabs at the bottom of the page, the user can move from page to page. Smaller boxes at the bottom of the window 200 give other information such as the total number of lots 230, total number of wafers 231 and total number of defects 232. A toolbar 240 at the top of window 200 has various buttons providing further selections, such as accessing other bar charts, the wafermap of Figure 4A, or the flow chart generating window of Figure 5A; toolbar 240 further provides buttons for changing the magnification of the bar charts, or for producing the calendar of Figure 2B.

In order to produce the chart of Figure 2B, the software package for creating the bar charts 200 of Figure 2A includes a dialog box button 241 in the toolbar kit 240. Dialog box 250 appears including a string selection drop down list box 277 shown in Figure 2B. The user then selects the method of entering dates from the string selection box 277. When the user chooses that the dates are to be selected using calendar 270, a month and year for the calendar 270 are selected from additional drop down list boxes 272, 273. Alternatively, the customer may use the string selection box 277 to enter the production variables in another manner, such as from a list box of text string, e.g. "last ten days," "last ten lots," or input a range of dates from the keyboard in a blank field box. Figure 2B further shows that calendar display 270 includes a plurality of push buttons 271 for each day of calendar 270. The user selects one or more push buttons 271 on the calendar 270 to indicate one or more days over which the production data is displayed in the bar chart format. It is also possible to select times within a given calendar

date by the hour and minute with dialog boxes 274, 275, or page through the calendar month by month with arrow keys 276.

Referring now to Figure 3, the software package includes a toolbar kit function that lets the user place two or more bar charts 310, 320 side by side for comparison by splitting the window 300. For example, the user has the capability of comparing a bar chart 310 of the number of defects by layer number for a first production lot relative to a similar bar chart 320 for a second production lot. The bar charts 310, 320 can be created as embedded packaged objects which can be dragged and dropped into another window, or the bar charts 310, 320 can be stored as files. The packaged object may be represented as an icon or other graphic image. The various pages are color coded for easier visibility.

2. Accessing Production Maps

Referring now to Figure 4A, the software package of the present invention includes a utility for accessing and displaying a production wafermap 410 in display window 400. The user zooms in or out on the production map by selecting a plus or minus zoom button bar 430. Each of the particular defects is color coded to facilitate visual identification in accordance with a defect number scale 440 to the right of map 410. Side-by-side with the wafermap, are shown Type Pareto and Size Histogram charts 450, 460 imported from windows 310 and 320 of Figure 3.

The actual production map is compared to a production map in a wafermap gallery to filter out all patterns except for defects. The user drags and drops a packaged image of the production map 410 with the computer cursor into one of the maps in the wafermap gallery. The actual and ideal production maps in wafermap gallery represent the way a particular product would appear if the product has no defects. A map 480 of the product without defects is accessed by selecting the tab 470 at the bottom of window 400 to page into wafermap gallery window.

Figure 4B shows the production map 440 after it has been filtered by information in the wafermap gallery 470. Once the actual production map 410 is subtracted from the ideal production map of the wafermap gallery

470, only a map 460 of the defects remains. By combining the selection of production parameters with filtering from the wafermap gallery 470, the user identifies the defect pattern in a particular lot number or layer number.

5 3. Creating Flowchart Recipes

Referring to Figure 5, the invention further includes a program function for building a customized flowchart 510 according to a selected recipe of conditionals. A conditional in this context is defined as an event that occurs or does not occur depending upon another value. This value
10 may be one of the production parameters from production data or a calculated result of some previous production function. A conditional produces one or more branches in a flowchart depending upon this value. The recipe of conditionals is a series of linked conditionals that is created by the user according to a desired data analysis and response scheme,

15 The flowchart building program function operates in the manner of a "macro" in a computer by recording keystrokes in the casebook and playing back the keystrokes of the casebook upon command from the user. The "macro" type program function of the present invention remembers a series of production parameters specified by the user from the dialog boxes.

20 The casebook macro is subsequently converted into flowchart 510 by the software and displayed. Each of the selected production parameters and production functions is assigned a geometrical element in the flowchart depending upon its function, e.g. box, circle, diamond. The geometrical elements of the flowchart correspond to their standard definitions in
25 flowchart lexicography; a circle is an input-output function, a diamond is a branching conditional, and a box is a data calculation. As shown in Figure 5, the user can also create the flow chart by dropping and dragging a node 520 from a dialog box 530 and linking it to one of the previous nodes 540 in the flow chart 510. Whenever the user wishes to execute a particular
30 analytical scheme on the production data, he selects a particular flowchart 510 from the current page by pressing a particular button 550 on the toolbar 560. The recipe represented by the flowchart may be readily modified by the

user by re-recording the conditional selections, redisplaying a new flowchart, and reexecuting the new flowchart more directly via the node dialog box 530.

The user may wish, for example, to execute on a regular basis a flowchart 510 that produces the standard deviation of defects in the second layer of the product for the last ten days. This particular flowchart 510 has been previously created and recorded by the user. Each of the pages in the casebook has a corresponding graphical node symbol 541, 542, 543, 544, 545 and 546 which is displayed in the window 500 upon selection. After this flowchart 510 is created, it is displayed in a linked fashion in the display window 500 as shown in Figure 5. The user then generates one or a series of bar charts as shown in Figure 2A, or performs other functions by selecting this particular flowchart 510 in the display window 500.

An analytical subroutine represented by the flowchart may be automatically executed by the user on a regular basis. Generic groupings of production data such as "last ten days" or "last ten lots" are specified rather than dates certain. This analytical subroutine is then executed, say, every week or every two weeks, without operator intervention. Reports of one or more of these analytical schemes are then automatically generated at these times. The reports may be generated automatically either at the operator's computer, where they are stored in data files, or at a print station. The operator specifies the number of copies of these reports to generate, and the location that they are generated is selected via the network printer.

A plurality of these reports are automatically published on a preestablished regular basis without taking up the operator's time on every occasion. Copies are routed to selected individuals responsible for overseeing production. Alternatively, instead of printing these reports on a network printer, they can be routed on a regular basis in an electronic format through a network server. The network can be either a local area network or an Internet location on the World Wide Web. Electronic publishing has the advantages of conserving paper and reducing costs.

It is also possible to automatically utilize the results of the previous data analysis subroutine to conditionally change the scheme of the data analysis, and the subsequent publication of the reports. If a condition is

detected where the results of a particular analysis exceed a preset value, then other predetermined analyses are performed to more precisely identify the source of the defect. Effectively, a feedback loop is set up to maintain tight control between the inspection results and the analytical scheme.

5 When the results of an analysis exceed predetermined control levels, a notification may be sent to the engineer through e-mail or pager systems to take a specified remedial action. A report might also be sent back to the production line to automatically initiate a process change, e.g. across the Internet to launch a Java applet. Thus, the analytical tool of the present
10 invention is more than just a data gathering mechanism since process control can be implemented automatically via a feedback loop.

One application of the analysis routine of the present invention is to determine when a defect is an "adder," i.e. a defect that is not present on any previous layer. These adder defects are subtracted from the defects that
15 were present on the previous layers, and then classified by the Automatic Defect Control (ADC) program.

4. System Architecture

As shown in Figure 6A, the software system of the present invention
20 utilizes a three-tier architecture. The highest tier 610 is the Presentation Layer, the object-oriented Windows[®] environment consisting of the menus, buttons, dialog boxes, and charts that are visible to the user. The middle tier 620 is the CORBA (Common Object Request Architecture) IIOP (Inter-Internet Object Protocol) Middleware Layer, which distributes objects across
25 servers and the Internet. The lowest tier 630, the Storage Layer, is a database containing raw production data which is connected by the CORBA Layer to other servers and the Presentation Layer.

Figure 6B shows how the CORBA Middleware Layer interfaces with the Storage Layer 630 and the Namespace Service 640 to modify of the cluster
30 classifications. CORBA is a mechanism of distributing objects across microcomputers and operating systems. The CORBA Middleware Layer provides access to the Server for objects that can reside anywhere on the network so that new cluster classifications can be presented from the

Namespace Service 640 to the DataSet Service 650. One client in Ireland may, for example, discover a new method of clustering 660 or custom compilation 670 of the data in that location. This method may be relevant to the needs of a client in the United States. CORBA takes the Namespace
5 Service 640 from the client in Ireland and applies it to the DataSet Service 650 from the United States over the server without having to access the production database in the United States to perform the analysis.

Figure 6C shows how CORBA interfaces via the server from the client's personal computer 651 to other work stations and the Database Server 652.
10 The Application Server 653 stores and distributes all of the various calculation modules or objects. The Application Server 653 can be the same as the Database Server 652, it can be the Client workstation 651, or anyplace else on the network. The recipes are contained in the Application Server 653, and once the recipes are run, the results can be published at a
15 printer 654, WEB Server 655 or client personal computer 656.

Figure 7 shows a detail of the software architecture of the Presentation Layer, written in Visual C++ programming language. The Recipe Editor 710 permits the user to create a Recipe Framework 720. The recipe is applied to the DataSet Framework 730 through the Application Modules 740, and with
20 the Graphic User Interface (GUI) Framework 750, a Page 760 is produced in the Windows® format. The Recipe Framework 730 also interfaces with the Scheduler and Event Service 770, providing automatic execution and publication of production data analyses at future specified times.

Once the analytical routines are performed, the results are displayed
25 in a graphical representation such as a bar chart, graph or filtered production map. Each cluster classification is displayed in a color coded format for greater visibility. For example, a filtered production map is produced with scratches color coded in red, particles in blue, and pinholes in green. Or, bar charts are created and placed side by side on the computer
30 screen for comparison, with one bar chart representing the standard deviation of scratches in lot one for the last ten days color coded in red, and a second bar chart representing the standard deviation of pinholes in lot one for the last ten days color coded in green.

5. Extending Classifications

Optical pattern recognition systems exist to permit a computer to classify certain types of defects. For example, a defect such as a scratch exhibits certain familial similarities to other scratches, i.e. the defect is long and thin. By abstracting the characteristic similarities of this class of defects, the computer recognizes and classifies based upon the abstracted characteristics. However, the various classes of defects that prior art pattern recognition systems can identify is typically fixed by the particular software program. It thus becomes desirable to be able to add new defect classifications when they become identifiable.

Defects often are present in groups or clusters since the causative factor leading to the defect is repeated in a general area on the wafer. These defect groups are referred to as cluster classifications. The optical pattern recognition system, working with a neural network computer, is capable of identifying these cluster classifications in a production map. Each cluster classification represents a filter that makes visible only the defect of that cluster classification. A plurality of defect filters may be applied in sequence so that only the applied combination of defects are operated on by a particular data analytical routine.

The present invention includes a software procedure for integrating and extending attributes of new cluster classifications into a namespace for an analysis routine. In this context, a namespace is similar to a filename but contains additional executable attributes that tell the operating system to perform predetermined operations because of the namespace extensions. An example of a namespace with attributes is a Word® file with a ".doc" extension. The ".doc" extension signals the operating system to do something when the user clicks directly on the filename, specifically to open the document in Word.® Otherwise, the user must first start the Word® word processing program, then click on the Open File list box on the title bar, then select the proper drive, and finally use the scroll bar to find and click on the correct file. The ".doc" extension is an attribute that tells the

computer to perform these tasks when the filename is selected, considerably simplifying the execution. The namespace is a collection of files represented by the filename including associated executable or other file types, these files running when the filename is selected.

5 Figure 8A shows the essential format of the namespace datafile. Cluster classification 800 has a namespace 801 with a filename 802 and an attribute 803. Attribute 803 has a definition 804 consisting of a number of executable and other type files 805. In this instance, files 805 cause the operating system to look for other files 805 that retrieve procedures for
10 executing an optical pattern recognition, filtering and display of data analysis in charts.

 In the present invention, the attributes of the cluster classification namespace include associated executable utilities for semiconductor defect analysis. Examples of attributes contained by this namespace include (1)
15 the series of commands associated with executing the optical pattern recognition program performed by the neural network, (2) the series of commands associated with performing an analytical scheme on the data collected from production lots, layers, dates, etc., and (3) the series of commands associated with importing and running a data analytical scheme
20 that is stored elsewhere on a network (e.g., via the Internet at a foreign subsidiary manufacturing plant or an intranet from another user's hard drive).

 Referring to Figure 8B, the extension is applied to cluster classifications of semiconductor defects. The program contains a table
25 namespace 810 of defect cluster classifications 811. The attributes of this namespace 810 include one or more executable or other type files telling the computer what to do when the cluster classification namespace 811 is click-selected with the computer mouse. These cluster classes 811 include such semiconductor defects 812 as a scratch, particle, blowout, pinhole, etc., or
30 the defects may be classified by size. When a scratch cluster classification namespace 812 is selected by the user with the cursor, the attributes of the scratch namespace 812 are executed to start the neural network pattern

identification procedure, to retrieve the data analysis parameters of the data analytical scheme, to execute this data analysis scheme, and to import any program or data files across a network as required.

Figure 8B further illustrates a list box 820 for adding cluster
5 classifications to the table namespace 810. The list box 820 is a drop down list box with a plurality of prospective cluster classifications 821. The user clicks on the drop down list box 820 with the computer cursor 822 to select one of these cluster classifications 821, and this classification 821 is added to the left column 811 of the table namespace 810. The new cluster
10 classification 821 is preferably moved from the drop down list box 820 to the table namespace 810 by dragging and dropping the new cluster classification 821 with cursor 822. The user thereby has a greater degree of flexibility to insert new attributes 803 into the namespace 810 than would be possible with a static software package. Upon execution of the cluster classification
15 821, the user filters the production map so that only the scratch data remains. Analysis of the scratch data can then be readily performed.

Figure 8C is a block diagram of the functional parts of the cluster class extending system of the present invention. Each of Clients A, B and C (831, 832, 833) are connected to a presentation layer (834, 835, 836) for their
20 respective computers. Clients A and B are in the United States, while Client C is in Ireland. Each of Clients A, B and C have a namespace 837 containing a filename 838 and an attribute 839 in each of their relational databases 840, 841, with Clients A and B sharing a relational database 840 containing definition 842 of the cluster classification filenames 838 and
25 attributes 839. A middleware layer 843 connects Client C to Clients A and B over the Internet. Client C is Ireland has his own relational database 841 containing namespace 837 and definition 842, but Client C also has namespace 844 with filename 845 and attribute 846 with definition 847. Client C has determined that namespace 844 is an especially good cluster
30 classification for analysis of his semiconductor data, and Clients A and B in the United States wish to acquire it. Client C therewith sends namespace 844 to Clients A and B.

Figure 8D is a flowchart showing the what happens after Clients A and B initiate a request for namespace 844 (step 850) and Client C sends the namespace 844 to Clients A and B (step 851). Client A then drags and drops the new cluster classification 821 representing namespace 844 into his table namespace 810 (step 852). Client A executes namespace 844 by selecting cluster classification 821 (step 853). Namespace 844 has an attribute 846. Execution of namespace 844 with attribute 846 causes Client A's computer to request transmission of namespace definition 847 across the Internet (step 854) via middleware layer 843. Client C's computer sends this definition 847, which resides in Client C's relational database 841 to Client A (step 855) via middleware layer 843. Thereupon, definition 847 is placed in relational database 840 shared by Clients A and B (step 856). Attribute 846 also contains an extension that identifies the presence of the particular cluster classification 821 with an optical pattern recognition device and a neural network computer (step 857) and executes a particular filtering routine on the production data (step 858) to produce a data analysis chart (step 859).

The invention has been described above according to a preferred embodiment, but those having ordinary skill in the art should recognize that changes, substitutions and variations can be readily made without departing from the spirit of the invention. Accordingly, the invention is only construed to be limited by the claims set forth below.

We claim:

1. A method of analyzing semiconductor production data, comprising the steps of:
 - 5 establishing a table namespace of old semiconductor cluster classifications having first attributes;
selecting a new semiconductor cluster classification having second attributes from a list box;
adding said selected new classification to said table namespace;
 - 10 choosing said new classification from said table namespace;
executing an analytical routine on data from said new classification with said second attributes.
2. The method of claim 1, said executing step further comprises at least
 - 15 one of:
importing a namespace containing said second attributes for said new classification;
performing a pattern recognition function for said new classification;
and
 - 20 generating a graphical representation of results of said analytical routine on said new classifications with said second attributes.
3. The method of claim 2, said importing step further comprising:
importing said namespace across the Internet or a local network.
- 25 4. The method of claim 1, further comprising:
displaying a plurality of graphical representations of said old and new classifications side by side.
- 30 5. The method of claim 1, wherein:
said executing of said analytical routine includes generating information about cluster classifications of defects.

6. The method of claim 5, wherein:

said generating of information about cluster classifications of defects includes generating information about scratches, particles, pinholes and blowouts.

5

7. The method of claim 1, further comprising the step of:
importing said new classification across a server.

8. The method of claim 1, further comprising the step of:
importing said new classification across the Internet.

10

9. A system for analyzing semiconductor production data, comprising:
a table namespace of old semiconductor cluster classifications having first attributes;

15

a list box containing a plurality of new semiconductor cluster classifications having second attributes;

an analytical routine for execution on data from either said old or new classifications with said first or second attributes.

10. The system of claim 9, said analytical routine further comprises at least one of:

an importation of a namespace having said second attributes for said new classification;

a pattern recognition function for said new classification; and

a graphical representation of results of said analytical routine on said old and new classifications.

25

11. The system of claim 10, wherein:

said namespace is imported over the Internet or a local network.

30

12. The system of claim 9, wherein:
said analytical routine is one of a number, mean, or standard
deviation of defects over a particular lot identification, layer number or time
period.

5

13. The system of claim 9, further comprising:
a plurality of graphical representations of said old and new
classifications displayed side by side.

10

14. The system of claim 9, wherein:
said analytical routine generates information about cluster
classifications of defects.

15

15. The system of claim 14, wherein:
said cluster classifications of defects include scratches, particles,
pinholes and blowouts.

20

16. A system for analyzing semiconductor production data, comprising:
table namespace means for storing and displaying old semiconductor
cluster classifications having first attributes;
list box means for storing and displaying a plurality of new
semiconductor cluster classifications having second attributes;
analytical routine means for execution on data from either said old or
new classifications with said first or second attributes.

25

17. The system of claim 16, said analytical routine further comprises at
least one of:

means for importing a namespace having said second attributes for
said new classification;

30

means for performing pattern recognition for said new classification;
and

means for graphically representing results of said analytical routine on said old and new classifications.

18. The system of claim 17, wherein:

5 means for importing said namespace over the Internet or a local network.

19. The system of claim 16, wherein:

10 said analytical routine means executes an operation of number, mean, or standard deviation of defects over a particular lot identification, layer number or time period.

20. The system of claim 16, further comprising:

15 means for displaying side by side a plurality of said old and new classifications.

21. The system of claim 16, wherein:

20 said analytical routine means generates information about cluster classifications of defects.

22. The system of claim 20, wherein:

said cluster classifications of defects include scratches, particles, pinholes and blowouts.

23. The system of claim 16, further comprising:

25 means for importing said new classification across a server.

24. The system of claim 16, further comprising:

means for importing said new classification across the Internet.

FIG. 1

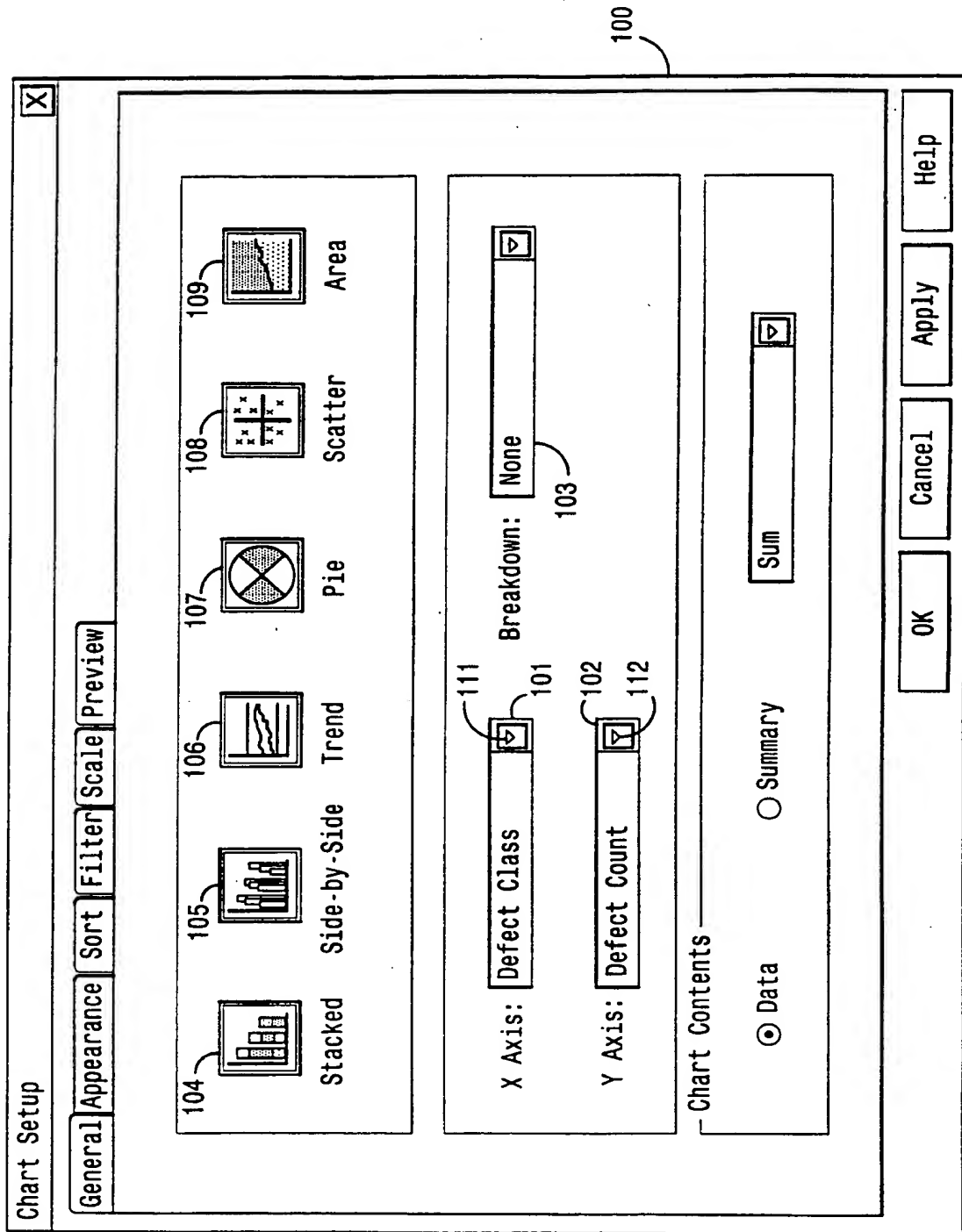


FIG. 2A

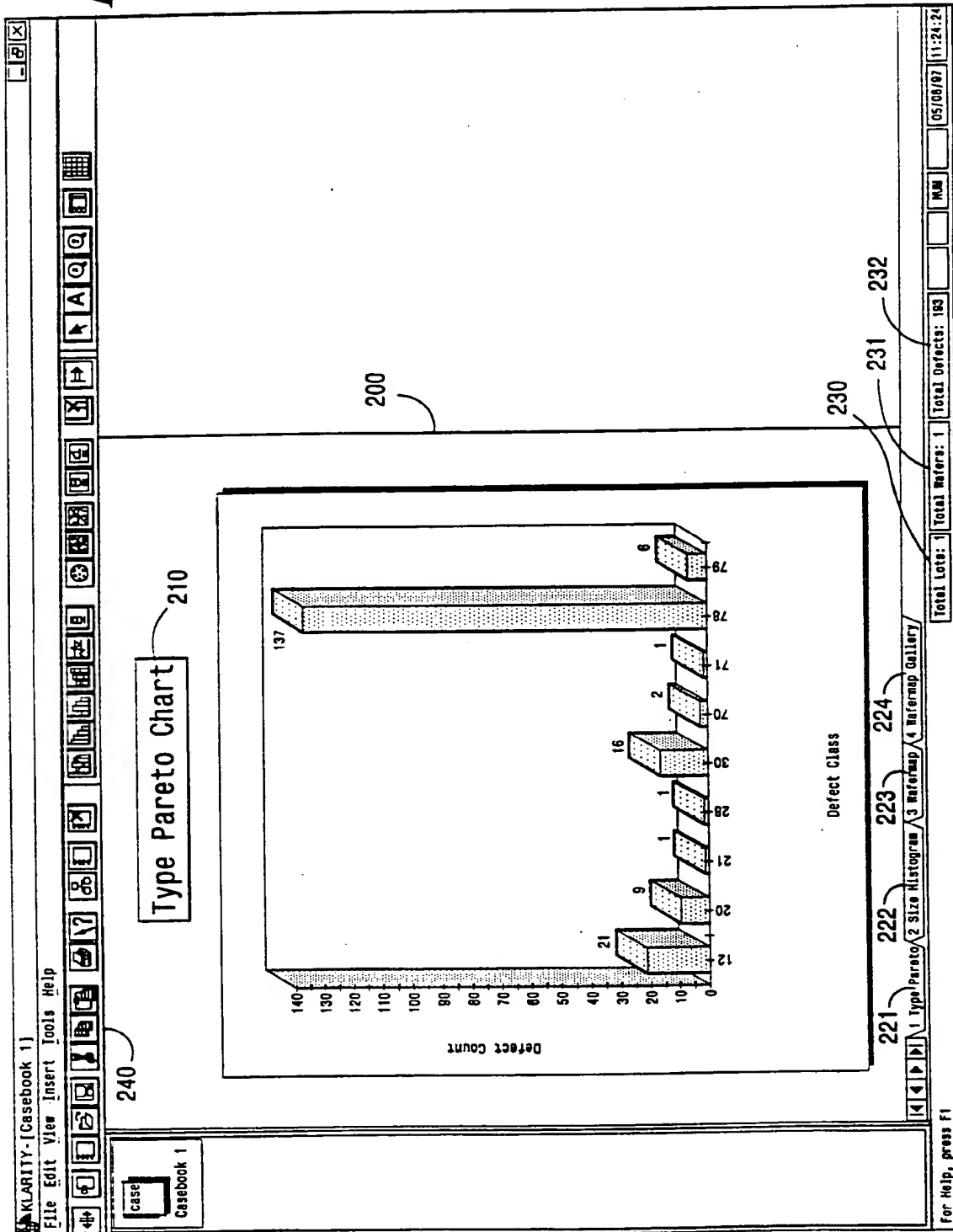


FIG. 2B

Dialog

String Selection USING CALENDAR

Between: Hour Min And: Hour Min

Current Month May Current Year 1997

BWD FWD

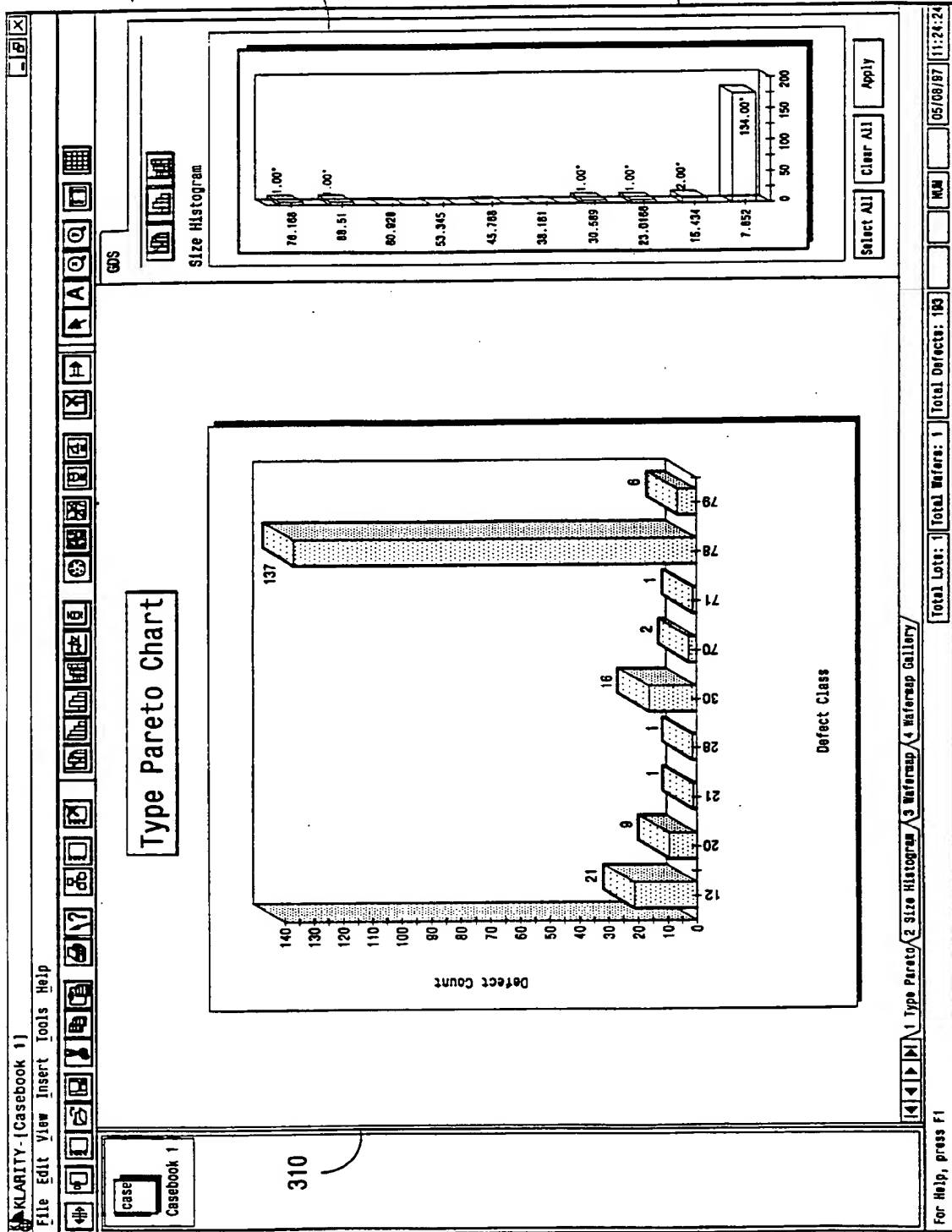
May 1997

OK Cancel

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

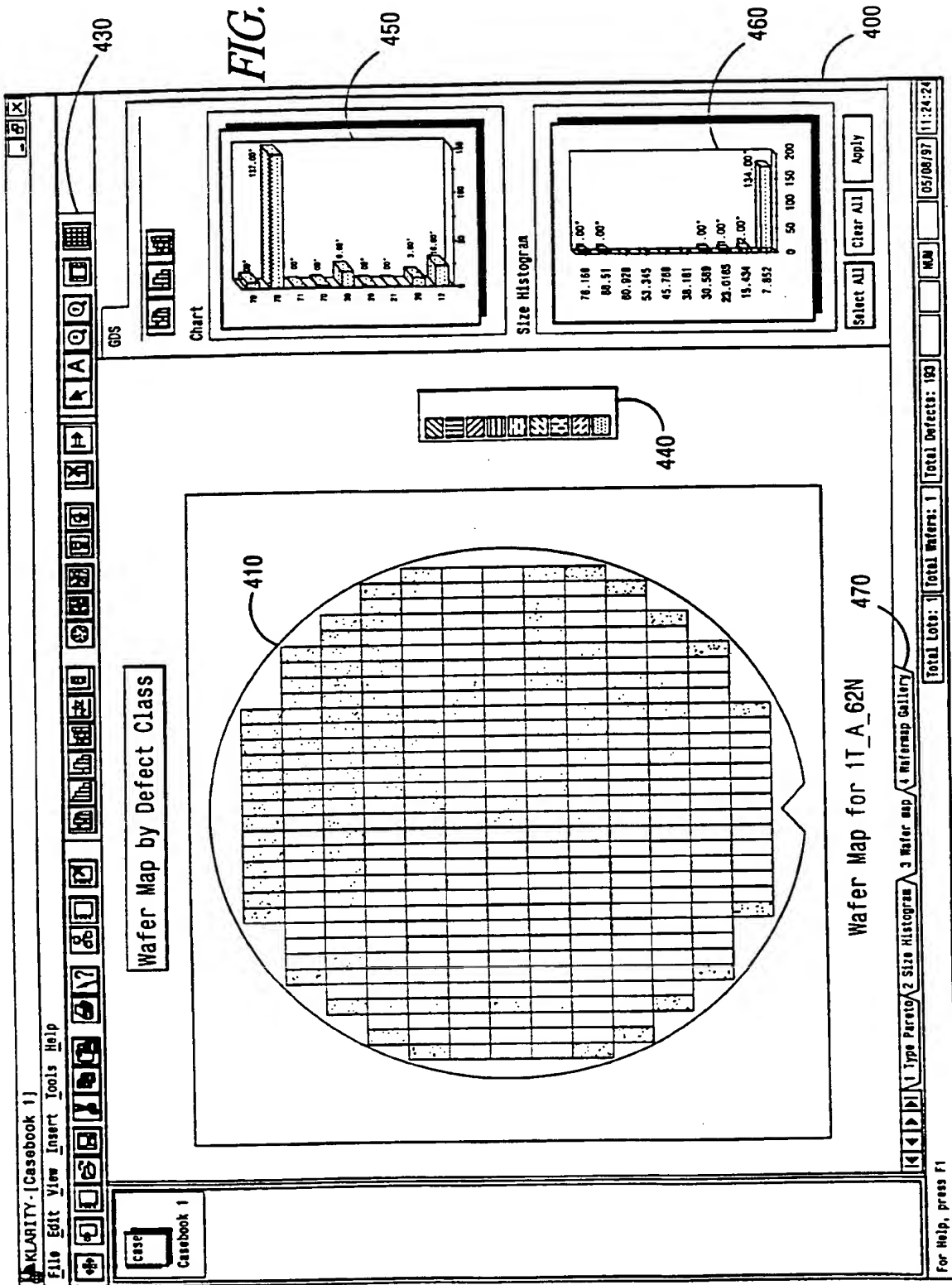
4/15

FIG. 3

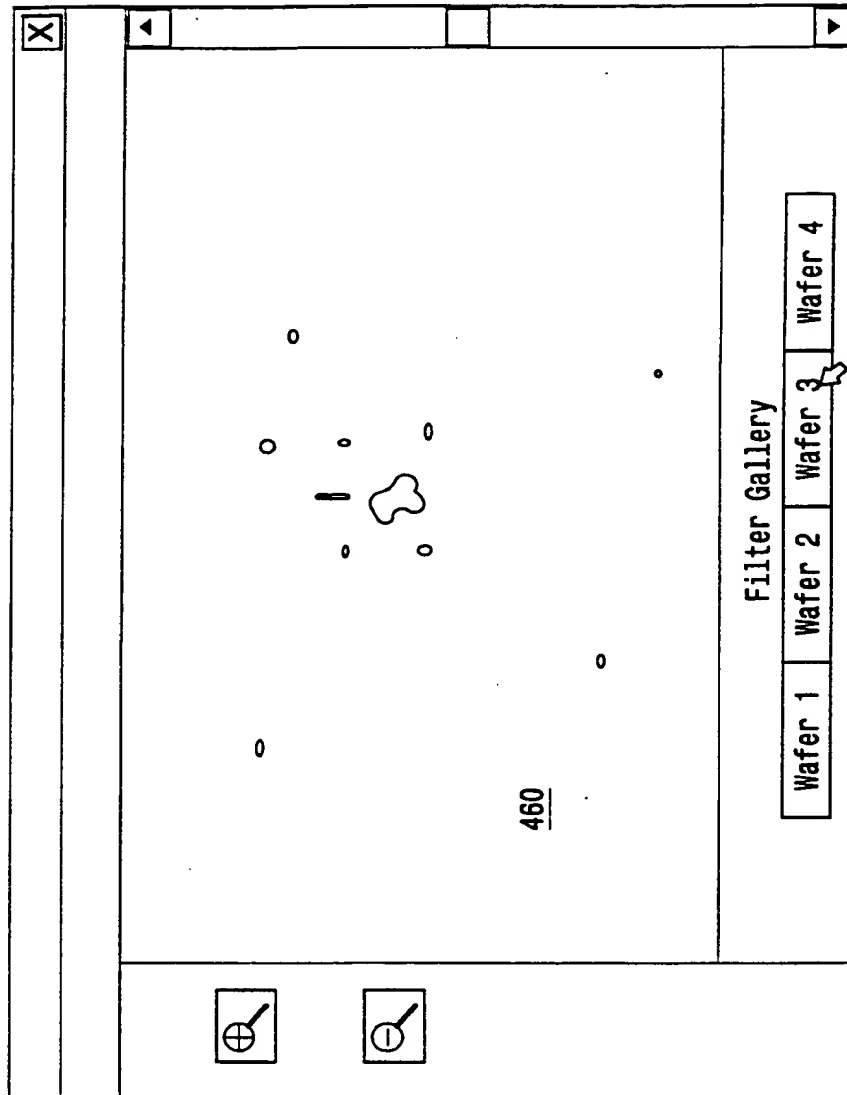


5/15

FIG. 4A

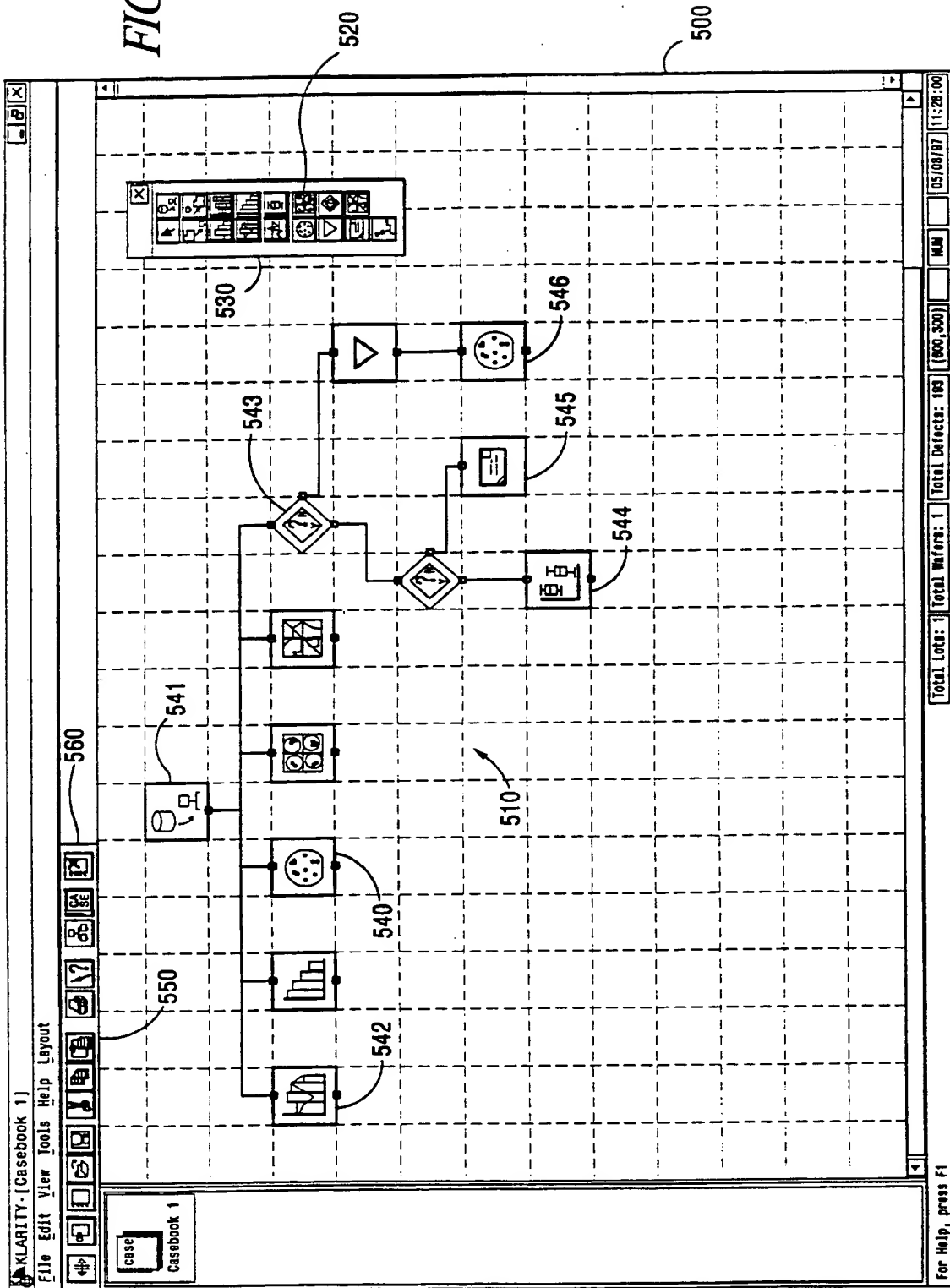


6/15

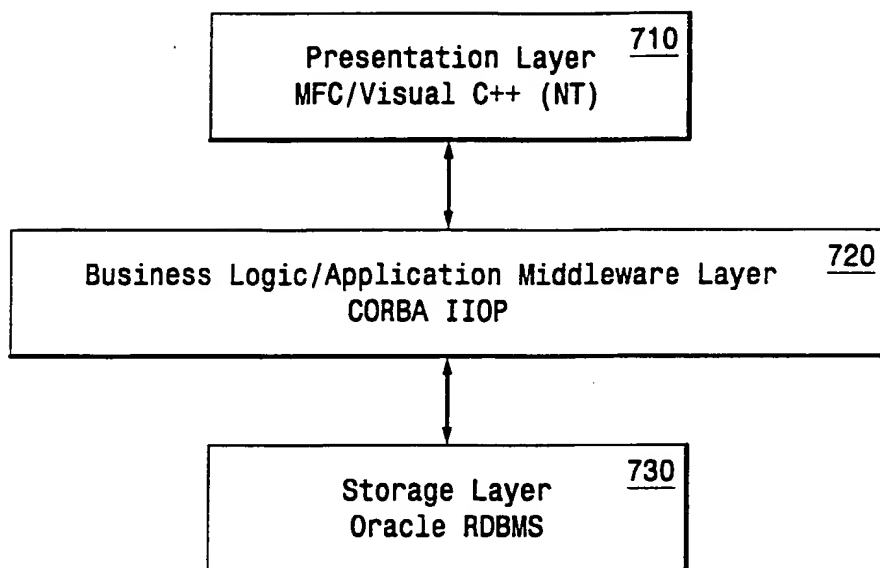
**FIG. 4B**

7/15

FIG. 5



8/15

*FIG. 6A*

9/15

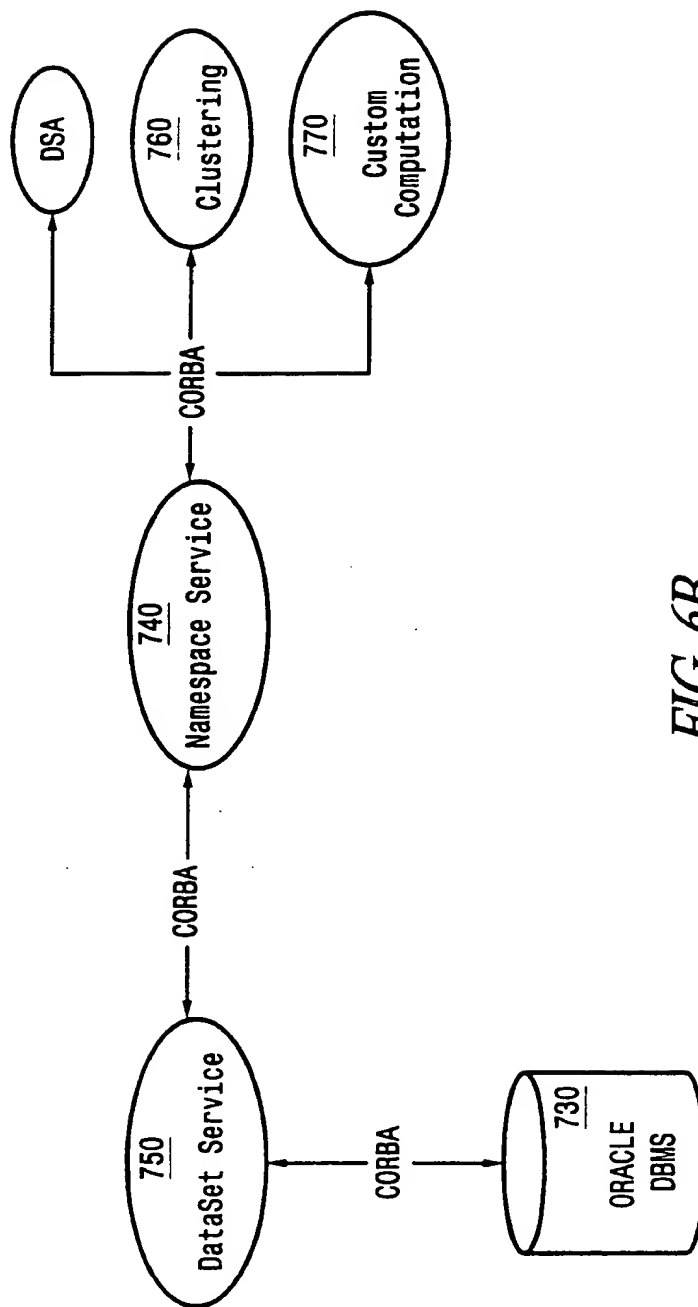
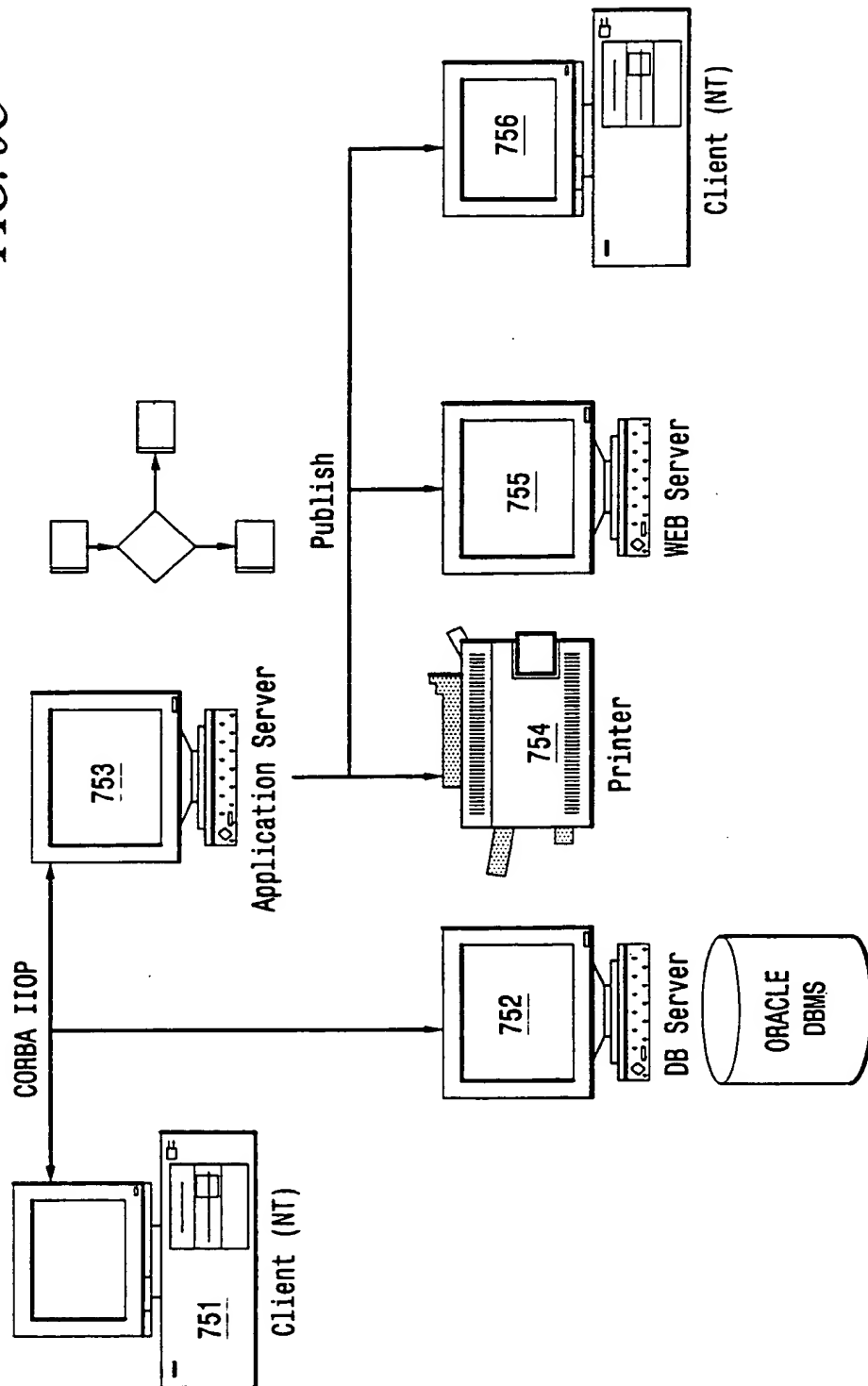


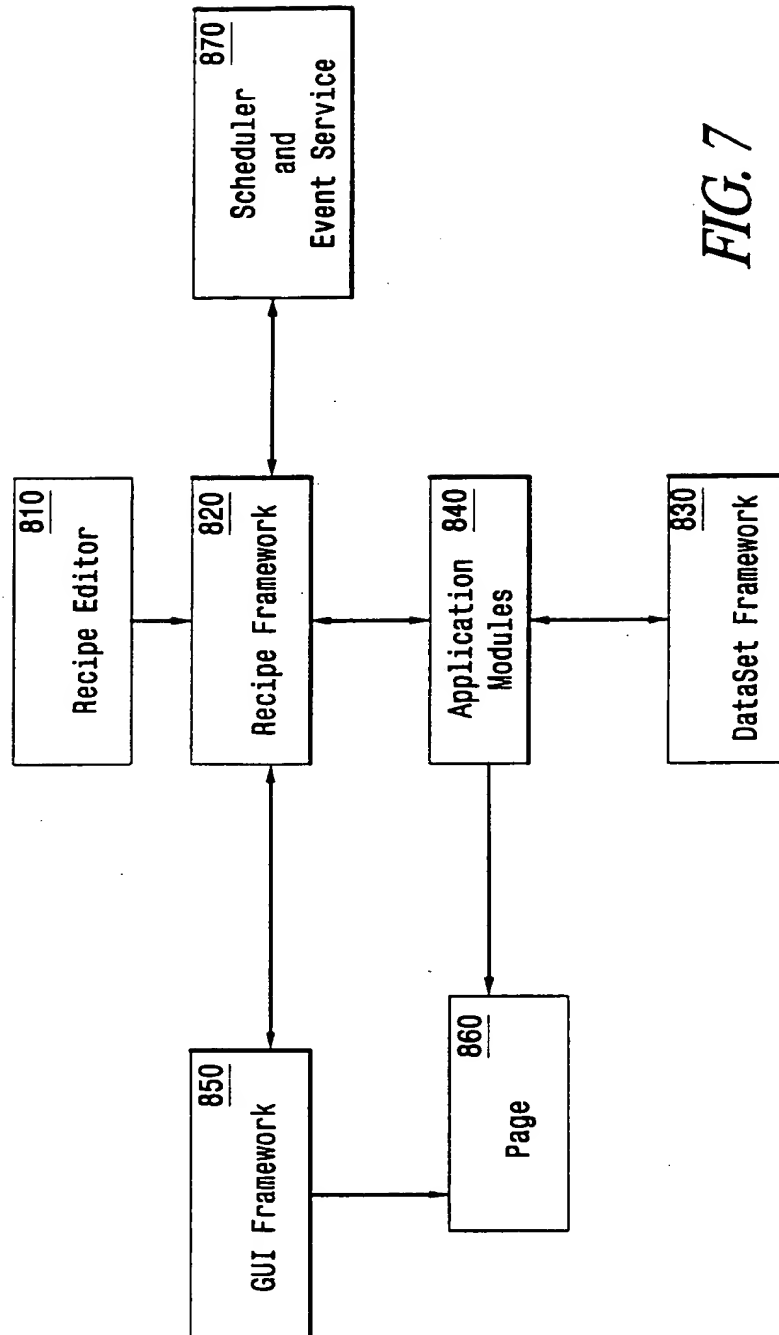
FIG. 6B

10/15

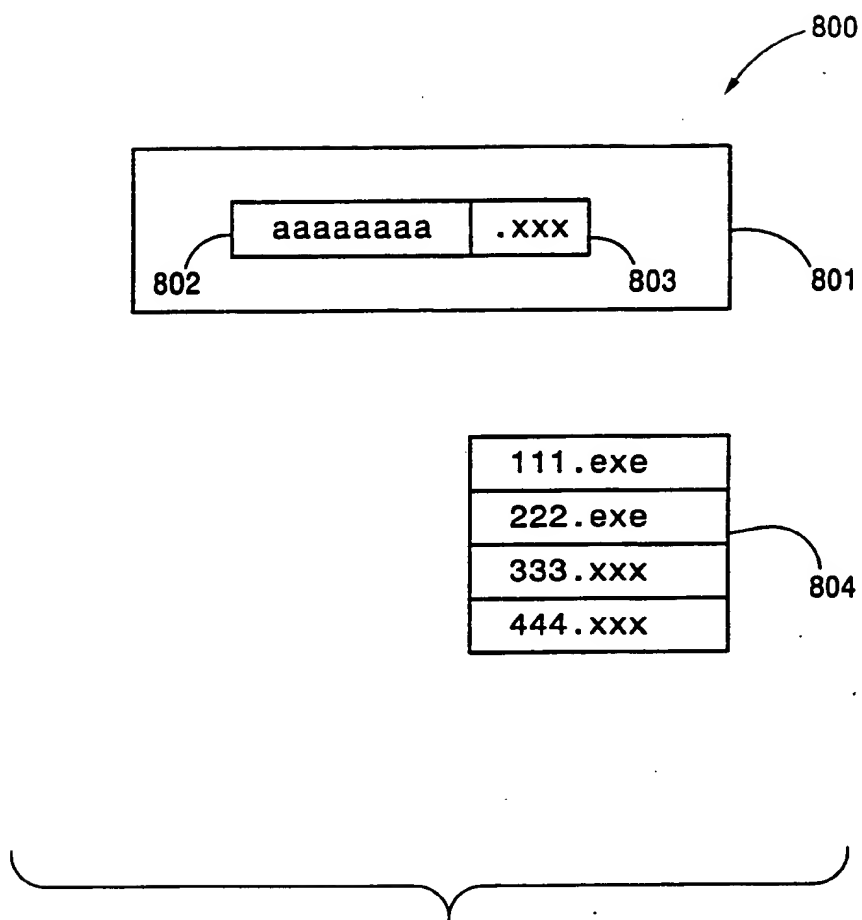
FIG. 6C



11/15

*FIG. 7*

12/15

*FIG. 8A*

13/15

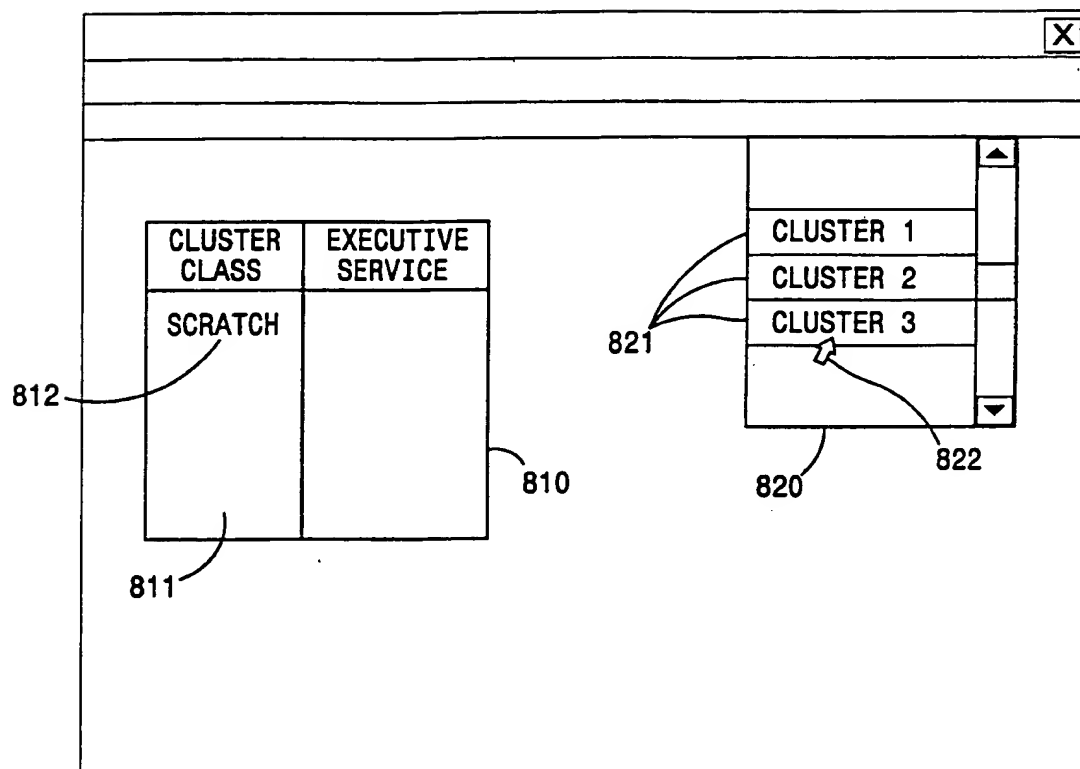
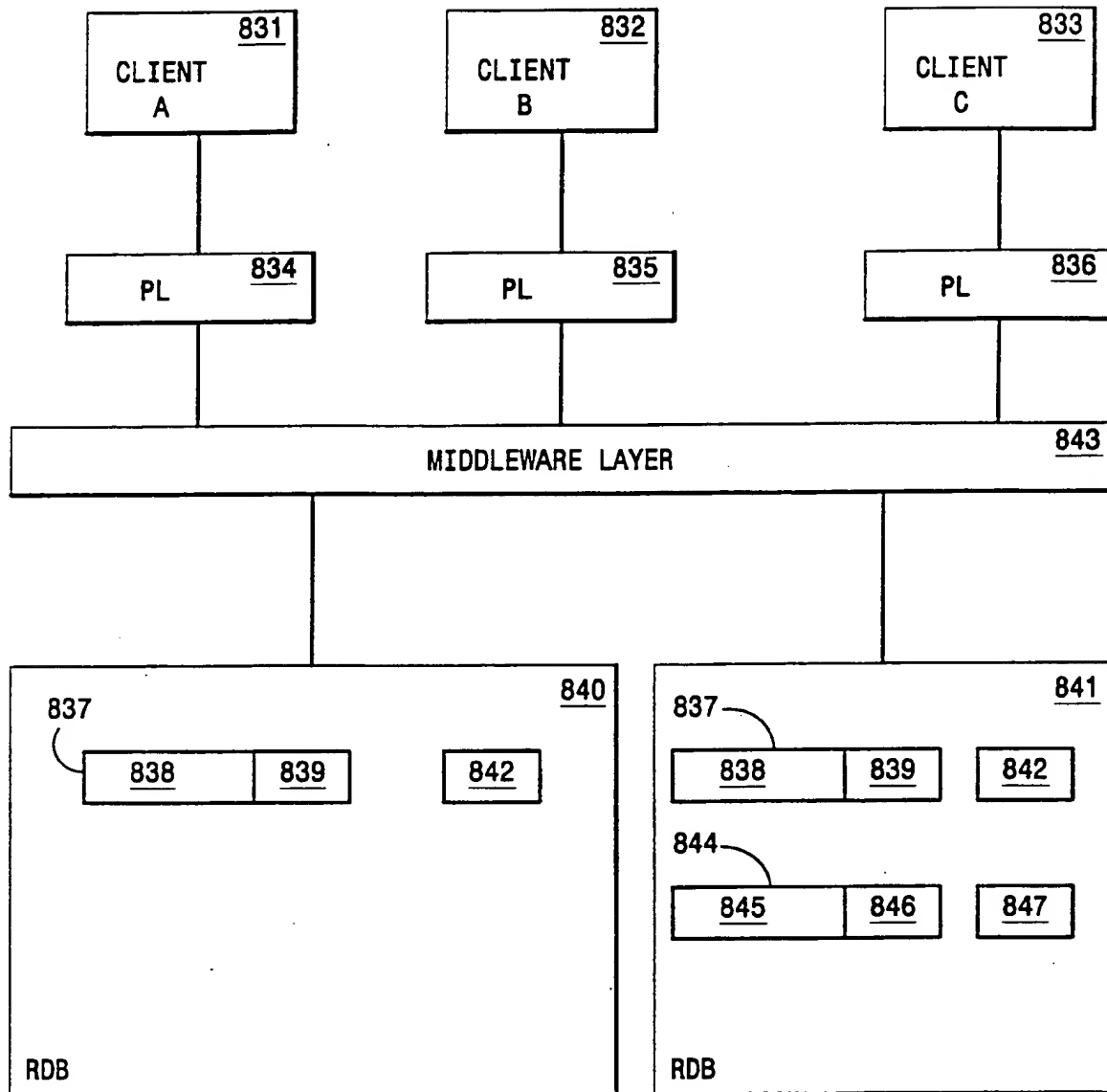
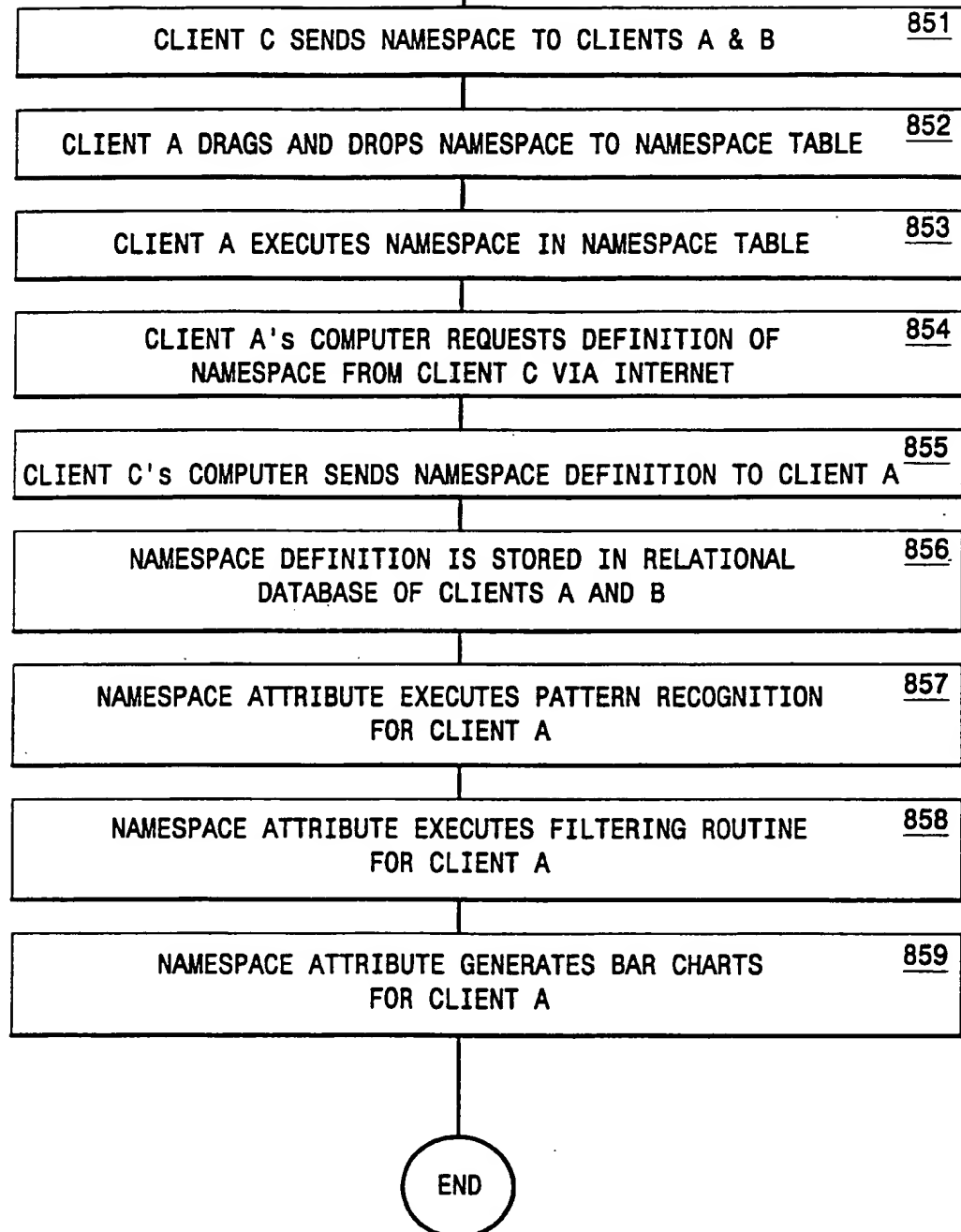


FIG. 8B

14/15

*FIG. 8C*

15/15

*FIG. 8D*

INTERNATIONAL SEARCH REPORT

International application No
PCT/US98/22735

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G06F 17/00, 17/50

US CL : 364/488, 489, 490, 491, 468.17; 395/161

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 364/488, 489, 490, 491, 468.17; 395/161

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,226,118 A (BAKER et al) 06 July 1993, abstract, col 1, lines 5-9, col 1, lines 39-53, col 2, lines 31-60, col 3, lines 13-48, figures 1, 7, 23-26.	1-24
Y, P	US 5,761,064 A (LA et al) 02 June 1998, abstract, col 1, lines 8-11, col 1, lines 23-43, col 2, lines 6-65, col 3, lines 23-67, col 5, col 15-30, col 6, lines 26-67, col 7, lines 13-65, claims 1-40, figures 1-10.	1-24
A, P	US 5,802,523 A (JASUJA et al) 01 September 1998, see entire Doc.	1-24
A, P	US 5,675,752 A (SCOTT et al) 07 October 1997, see entire Doc.	1-24

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A document defining the general state of the art which is not considered to be of particular relevance	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
E earlier document published on or after the international filing date	*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*A* document member of the same patent family
U document referring to an oral disclosure, use, exhibition or other means	
P document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 21 DECEMBER 1998	Date of mailing of the international search report 03 MAR 1999
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer KEVIN TESKA Telephone No. (703) 305-9704 Joni Hill

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/22735

B. FIELDS SEARCHED

Electronic data bases consulted (Name of data base and where practicable terms used):

APS, IEEE, GPIC

Search terms: dialog box## or list box##, attribute#, analytical tool#, database or excel, oracle version 7?, semiconductor, production data, defect#, classification#